Problem Set #3

Greg Plaxton

February 25, 2019

Your solutions are required to be typeset using LaTeX, and to be turned in electronically using Canvas. As discussed on the course syllabus, there are two deadlines for this homework. The first deadline is 8pm on Friday, March 8. The second deadline is 8pm on Sunday, March 10. Submissions received before the first deadline will not be charged any slip days. No submissions will be allowed after the second deadline. See the syllabus for further details regarding the slip day policy.

1. In this problem you will prove the working-set theorem for splay trees that is stated on the associated set of lecture slides. As in the proof of the static optimality theorem presented in the lecture, you will prove the working-set theorem by choosing a suitable weight assignment for the keys. However, you will not use a static weight assignment. Instead, you will use a move-to-front (MTF) list to determine the weights before the first access and after each subsequent access. Initially, the MTF list contains the n keys in some arbitrary order. When a key x in position i of the MTF list is accessed, where $1 \leq i \leq n$, we update the MTF list as follows: The key x is moved to position 1, and the key that was in position j before the access is moved to position $j+1$ for $1 \leq j < i$. At any given point in the access sequence, the state of the MTF list determines the weights assigned to the keys, as follows: The key in position i of the list is assigned a weight of i^{-2} for $1 \leq i \leq n$.

   (a) Prove that the weight reassignment procedure performed after each access does not increase the potential. Note: You should make use of the same potential function as in the proof of the static optimality theorem.

   (b) Prove that the maximum possible drop in potential over the entire access sequence is $O(n \log n)$.

   (c) Prove the working-set theorem.

2. Design a data structure that maintains a dynamic set S of integer pairs (x, y) such that (1) the usual dictionary operations Insert, Delete, and Find are supported in $O(\log |S|)$ time, and (2) the special operation Sum defined next is also supported in $O(\log |S|)$ time. The operation Sum takes as input two integers a and b such that

$a \leq b$, and returns the sum of the y-values of all pairs (x, y) in S such that $a \leq x \leq b$. Justify your answer.

3. Recall that in a Fibonacci heap, we perform a cascading cut whenever we would remove a second child from a non-root node. Suppose we change the definition of the data structure so that a cascading cut is only performed when we would remove a third child from a non-root node. For any nonnegative integer k, let a_k denote the minimum size of any subtree rooted at a node with k children.

   (a) Give a recurrence for computing the sequence of a_k values. Justify your answer.

   (b) Prove that there is a constant $c > 1$ such that $a_k \geq c^k$.

4. Problem 24–6, page 682.